

Due Date: October 8, 2003 #18

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:)

Inventor: Howard Justin Glaser et al.)

Serial #: 09/162,685)

Filed: September 29, 1998)

Title: HTML MAPPING SUBSTITUTION)
GRAPHICAL USER INTERFACE FOR)
DISPLAY OF ELEMENTS MAPPED)
TO HTML FILES)

Examiner: William L. Bashore

Group Art Unit: 2176

Appeal No.: _____

SUPPLEMENTAL BRIEF OF APPELLANTSBoard of Patent Appeals and Interferences
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450**RECEIVED**

OCT 14 2003

Technology Center 2100

BOARD OF PATENT
APPEALS &
INTERFERENCES
OCT - 8 2003

Dear Sir:

In accordance with 37 CFR §1.192, Appellants hereby request reinstatement of the Appeal (pursuant to the Appeal Brief filed by Appellants on February 28, 2003) and submit the Appellants' Supplemental Brief on Appeal from the rejection in the above-identified application, in triplicate, as set forth in the Office Action dated July 8, 2003.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

An Appeal Brief was earlier submitted by Appellants on February 28, 2003. In response, prosecution was reopened resulting in the Office Action mailed on July 8, 2003. In response to the

July 8th Office Action, this Supplemental Brief is accompanied by a request to reinstate the Appeal.

III. STATUS OF CLAIMS

Claims 1-34 are pending in the application. Pursuant to the earlier filed Appeal Brief, all prior objections and rejections were withdrawn.

Claims 1-2, 4, 11-13, 15, 22-24, 27, and 33-34 are currently rejected under 35 U.S.C. 103(a) as being unpatentable over screen shots from HTMLed (HTMLed).

Claims 3, 5, 14, 16, 25, and 27 are currently rejected under 35 U.S.C. 103(a) as being unpatentable over screen shots from HTMLed in view of screen shots from Nano WebEditor (Namo).

Claims 6-8, 17-19, and 28-30 are currently rejected under 35 U.S.C. 103(a) as being unpatentable over screen shots from HTMLed in view of Arora et. al., U.S. Patent No. 5,911, 145 (Arora).

Claims 9-10, 20-21, and 31-32 are currently rejected under 35 U.S.C. 103(a) as being unpatentable over screen shots from HTMLed in view of Lisle et al., U.S. Patent No. 6,069,630 (Lisle).

IV. STATUS OF AMENDMENTS

No amendments to the claims have been made subsequent to the Office Action.

V. SUMMARY OF THE INVENTION

Appellants' invention, as recited in independent claims 1, 12, 23, and 34, is generally directed to a displaying a relationship between an HTML file and an element from a form (see page 3, lines 16-18). Specifically, a mapping/relationship is displayed in a graphical user interface. The mapping is between an HTML file and an element from a form, wherein the element was transferred from the form to an HTML page. (See page 5, lines 8-12).

The graphical user interface indicates the relationship between the element, the form that the element was transferred from, and an HTML file that the element was transferred to (see page 5, lines 8-27). Such claim language (in the independent claims) illustrates that the user may easily view

the relationships of objects involved in the creation of a web page in a graphical user interface. Such viewing capability may further allow a user to easily manipulate a page and various elements in a page using a tool for building an HTML page (associated with an HTML file). Further, as described in the specification, since the form contains one or more elements, and many instances of an element may be used in various HTML pages, only one copy of the form (that includes the element) needs to be retrieved locally (see page 15, lines 1-5). Accordingly, processing and transfer time are optimized.

Once the information is read from the project file that contains the relationship between the element and a particular HTML file, the information is processed to obtain a mapping from the element in the form to the HTML file (see page 31, line 12 – page 32, line 12). Once obtained, the mapping is displayed in a graphical user interface (see page 32, lines 15-16). The graphical user interface allows the user to identify the element in the HTML file, the form where the element was transferred from, and the HTML file where the element was transferred to.

Dependent claims 2, 13, and 24 provide a timing element that specifies when the information in the independent claims is generated. Specifically, the information is generated when the element is transferred from the HTML page associated with the HTML file (see page 5, lines 15-17).

Dependent claims 3, 14, and 25 provide the element from the independent claims is selected from a group that comprises a visual control and a non-visual. Further, the non-visual control is selected from a group comprising a button, a picklist, and a data entry box. (See page 12, line 15 – page 14, line 3)

Dependent claims 4, 15, and 26 specify that the information comprises an element name and an HTML file name (see page 32, lines 3-5).

Dependent claims 5, 16, and 27 depend on claims 4, 15, and 26 and further add the limitation that the information also comprises a form name (see page 32, lines 3-5).

Dependent claims 6, 17, and 28 provide details regarding how to display the mapping. Specifically, these claims provide that the mapping is displayed by presenting an element name and an HTML file name in a row of a table (see Fig. 22).

Dependent claims 7, 18, and 29 further elaborate on the table of claims 6, 17, and 28. Specifically, these claims indicate that the table has cells that are defined by row and column and the mapping is entered into the cell (see Fig. 22).

Dependent claims 8, 19, and 30 provide alternative details regarding how to display the mapping. Specifically, these claims provide that the mapping is displayed by presenting an element name and the HTML file name in a column of a table (see Fig. 22 and page 33, line 19 – page 34, line 11).

Dependent claims 9, 20, and 31 provide the additional capability to flag a mapping (between the element and HTML file) that is invalid (see page 32, lines 14-15).

Dependent claims 10, 21, and 32 further elaborate the flagging of claims 9, 20, and 31. Specifically, a project file is read and the HTML file name is extracted from the project file. Thereafter, the HTML file name is used to search for the HTML file. If the HTML file is not found, the mapping is flagged as invalid (see page 33, lines 10-18).

Dependent claims 11, 22, and 33 provide for modifying the mapping. Specifically, a modified mapping is accepted and then stored in a project file (see page 32, lines 17-23).

VI. ISSUES PRESENTED FOR REVIEW

Whether claims 1-2, 4, 11-13, 15, 22-24, 27, and 33-34 are unpatentable under 35 U.S.C. 103(a) as being rendered obvious over screen shots from HTMLcd.

Whether claims 3, 5, 14, 16, 25, and 27 are unpatentable under 35 U.S.C. 103(a) as being rendered obvious over screen shots from HTMLcd in view of screen shots from Namu.

Whether claims 6-8, 17-19, and 28-30 are unpatentable under 35 U.S.C. 103(a) as being rendered obvious over screen shots from HTMLcd in view of Arora.

Whether claims 9-10, 20-21, and 31-32 are unpatentable under 35 U.S.C. 103(a) as being rendered obvious over screen shots from HTMLcd in view of Lisle.

VII. GROUPING OF CLAIMS

The rejected claims do not stand or fall together. Each claim is independently patentable. Separate arguments for the patentability of each claim are provided below.

VIII. ARGUMENTS

A. The Independent Claims are Patentable Over the Cited Art

The Office Action rejects claim 1 as follows:

In regard to independent claim 1, HTMLed teaches an HTML editor comprising a Form Designer for creating/editing a form to be integrated within an HTML page (HTMLed page 3). The Form Designer allows a user to initially drag and drop various form elements onto a designer field as shown by the example created form (HTMLed page 5). After pressing "OK", the code responsible for creating the form is transferred (associated) with a newly created HTML file (sample.htm) (HTMLed page 6). When sample.htm is reopened and/or previewed in Netscape via "Test with Netscape" option (HTMLed page 7-8), the form elements described by Form Designer are defined in sample.htm, which in turn are defined in the page produced in Netscape from said file, therefore displaying a relationship by at least viewing the HTML code (compare with claim 1 "*A method of displaying a relationship between an HTML file and an element from a form, wherein the element is in an HTML page, comprising:*").

HTMLed teaches a file (sample.htm) containing form control code, said sample.htm file can be reasonably interpreted by the skilled artisan as a project file, said sample.htm file containing form code information defining an association between the previously transferred form elements and the page rendered from said sample.htm file (HTMLed page 6; compare with claim 1 "*reading information from a project file, the information comprising a relationship between the element that has been transferred from the form to the HTML page and the HTML file associated with the HTML page;*").

HTMLed teaches creation of a customized form using a FORM Designer window, said window showing a spatial mapping of form elements (HTMLed page 5). Subsequent to pressing the "OK" button, the representative form code information is transferred (mapped) to an HTML file (sample.htm, see HTMLed page 6), the code also acting to preserve the spatial mapping of said elements (compare with claim 1 "*processing the information to map the element from the form to the HTML file;*").

HTMLed teaches display of a sample HTML file (sample.htm, see HTMLed page 6) comprising Form code, said form code showing an association between form elements (i.e., "Radio Button" is associated with the form), said form code showing a relationship to an HTML file (i.e., the form code is enclosed within sample.htm). The limitation of displaying a mapping would have been obvious to one of ordinary skill in the art at the time of the invention, in view of HTMLed, because of HTMLed's disclosure of sample.htm, and of Form Designer. Since the form elements are associated (related) to the displayed form code block, and said form code block is associated (related) to sample.htm, the preservation of spatial mapping of form elements (HTMLed page 5), provides a reasonable suggestion to the skilled artisan to display the above associations, relations, and preserved spatial mappings as a displayed mapping of relationships, providing the benefit of mapping for a user to better visualize form relationships (compare with claim 1 "*displaying the mapping on a graphical user interface that indicates the relationship between the element, the form, and the HTML file;*").

For reasons similar to those earlier addressed, Appellants respectfully traverse and appeal these rejections for one or more of the following reasons:

- (1) *None of the cited references teach, disclose, or suggest providing a mapping from an element transferred from a form to an HTML file associated with an HTML page;*

(2) *None of the cited references teach, disclose, or suggest displaying a mapping from an element to an HTML file in a graphical user interface that indicates the relationship between the element, the form, and the HTML file.*

As described above, the present claims provide for displaying (in a graphical user interface) a relationship (i.e., mapping) between an HTML file, an element from a form, and the form, wherein the element was transferred to an HTML page from the form. For example, a form can contain multiple elements. One element is transferred from the form to an HTML page (e.g., when creating the HTML page).

As claimed, the graphical user interface indicates the relationship between the element, the form that the element was transferred from, and an HTML file that the element was transferred to. Such claim language illustrates that the user may easily view the relationships of objects involved in the creation of a web page in a graphical user interface. Such viewing capability may further allow a user to easily manipulate a page and various elements in a page using a tool for building an HTML page (associated with an HTML file). Further, as described in the specification, since the form contains one or more elements, and many instances of an element may be used in various HTML pages, only one copy of the form (that includes the element) needs to be retrieved locally (see page 15, lines 1-5). Accordingly, processing and transfer time are optimized.

Once the information is read from the project file that contains the relationship between the element and a particular HTML file, the information is processed to obtain a mapping from the element in the form to the HTML file. Once obtained, the mapping is displayed in a graphical user interface. The graphical user interface allows the user to identify the element in the HTML file, the form where the element was transferred from, and the HTML file where the element was transferred to.

HTMLeled completely fails to teach any such relationship or display of such a relationship. In fact, the Office Action admits the lack of displaying the relationship by stating that display of a mapping would be "obvious" in view of HTMLeled (see page 4 of the Office Action).

The distinction between the terms "form" as used in the claims and "HTML Form" as used in HTMLeled must be understood to understand the differences between the reference and the claimed invention. The form as claimed provides a list of one or more elements that may be used in

an HTML page including an HTML Form. An HTML Form is a specific type of HTML document that is used to obtain information from an end-user. In other words, the HTML Form is a document (that is created by a programmer or web-site developer) that may be filled in (e.g., using radio buttons, text entry fields, list boxes, etc.) by an end-user and then submitted for processing (of the information obtained) by a script at the web-site.

To teach the relationship aspect as claimed, the Office Action relies on the sample.htm file and the Form code within the file (as displayed on page 5 of HTMLed). However, such screen shots of HTMLed do not even remotely suggest the mapping as claimed. The screen shots of HTMLed illustrate the use of a Form Designer that has various Form elements to create a new HTML Form (i.e., the sample.htm document) (see page 3 of HTMLed). In this regard, the Form Designer may be viewed as a form that is used to create an HTML Form.

The Office Action suggests that HTMLed page 5 teaches the Form Designer window showing a spatial mapping of form elements. However, the only thing that the Form Design window of page 5 shows is a list of possible Form elements that may be used (on the right side of the window) and the HTML Form that has been created using such elements (on the left hand side). In this regard, there is no mapping displayed or suggested. As described on page 3 of HTMLed, the user may merely select the desired Form element and drags it into the window that displays the currently designed HTML Form. As claimed, the relationship and mapping comprises the element, the form the element was transferred from, and the HTML file. HTMLed merely illustrates an HTML Form and a list of possible elements that can be placed into the HTML Form. Additionally, the sample.htm file shows the code used to create the HTML Form displayed on page 5.

Examining the HTML code on page 6 of HTMLed, the first line provides that the form input is given/processed (i.e., by a script) via the "POST" method. The "ENCTYPE" tag indicates how the form data is encoded. The next three lines indicate that there are three radio button elements in the form whose values are stored in the variables radio06, radio07, and radio08. Further, the text "Radio Button" is displayed after each one of the radio buttons. Additionally, the third radio button has a default value that is "checked". After the radio buttons, the code provides for displaying a text area that is stored in the variable text02. The next line in the code provides for displaying a "Submit" button that may be used for submitting the form to the script that processes

the input. The result of the form created by the sample.htm code of page 6 is displayed on page 8 of HTMLed.

As illustrated throughout HTMLed (and in the sample.htm code), there is no reference to the form or mapping to the Design Form whatsoever. Instead, the Design Form is merely used to create an HTML Form with standard HTML Form elements. Further, the standard HTML Form elements do not teach or display a relationship or mapping. The Office Action suggests that the Form code within sample.htm (on HTMLed page 6) teaches an association between Form elements and a rendered page (see page 3 of the Office Action). However, the HTMLed reference merely contains standard HTML Form elements (i.e., radio buttons) in an HTML Form as described above. Thus, contrary to that suggested in the Office Action, the reference merely teaches standard code (as illustrated in paragraph 3 of page 3 of HTMLed and as illustrated in the reference pages of "Teach Yourself Web Publishing with HTML in a Week" attached herein as Appendix B). Further, as set forth in the Office Action, the HTML Form code is merely transferred to the HTML file (sample.htm) when the "OK" button is selected. Contrary to that asserted in the Office Action, such a transfer is not equivalent to a mapping. Instead, the code is literally transferred into the HTML file.

Again, the Office Action provides (i.e., in paragraph 4 on page 3) that the HTML Form code shows "a relationship to an HTML file (i.e., the form code is enclosed within sample.htm)." However, such an assertion is not even remotely accurate. As described above, the HTML file merely contains standard HTML Form code that is part of an HTML Form. There is no mapping between an element, a form that the element was transferred from, and an HTML file (as claimed). In this regard, even if HTMLed shows a relationship between HTML Form elements and an HTML file, HTMLed would still fail to teach all of the elements of the relationship/mapping as claimed - (1) the element, (2) the form the element was transferred from, and (3) the HTML file. In fact, HTMLed does not even remotely refer to or suggest recording/storing the name or attributes of the particular Design Form that an element was retrieved/transferred from. Instead, the HTML Form code is merely placed in the sample.htm file.

The Office Action then continues and states (on page 4):

...Since the form elements are associated (related) to the displayed form code block, and said form code block is associated (related) to sample.htm, the preservation of spatial mapping of form elements

(HTMLcd page 5), provides a reasonable suggestion to the skilled artisan to display the above associations, relations, and preserved spatial mappings as a displayed mapping of relationships, providing the benefit of mapping for a user to better visualize form relationships.

Appellants submit that such an assertion is completely without merit. In this regard, the HTML Form elements are defined in the HTML file by the HTML Form code blocks. There is no reference, implicit or explicit, of the form that the element was transferred from whatsoever (as claimed). Further, HTMLcd does not display a mapping ("spatial" or otherwise) of form elements. In addition, the Office Action relies on the benefit of mapping for a user to better visualize form relationships to suggest the display of the claimed mapping/relationship. However, such a benefit or suggestion of such a benefit is completely absent from HTMLcd. In this regard, unlike the present invention, the prior art fails to recognize the benefit and possibility of creating a mapping and displaying such a mapping as presently claimed.

In view of the above, Appellants submit that it would not have been obvious to display the mapping as claimed. Further, the existence of such a mapping (for display or otherwise) is clearly absent from HTMLcd and the other references.

B. Dependent claims 2, 13, and 24 are Patentable over the Prior Art

Dependent claims 2, 13, and 24 provide a timing element that specifies when the information in the independent claims is generated. Specifically, the information is generated when the element is transferred from the HTML page associated with the HTML file. None of the cited references teach, disclose, or suggest such a timing element as claimed. The Office Action relies on HTMLcd wherein when the user presses the "OK" button, the HTML Form code is transferred to an HTML file. The Office Action submits that such a transfer is equivalent to a mapping. As described above, Appellants respectfully traverse such an assertion. The mapping that is created and displayed is very different from and is not equivalent to the mere transfer of code from one location to another location. As described above, the mapping is between the element, the form that the element was transferred from, and the HTML file that the element was transferred to. Such a mapping is not even remotely suggested or described in HTMLcd. Without describing the mapping, HTMLcd cannot possibly describe the timing for when the claimed mapping is generated.

Thus, Appellants submit that dependent claims 2, 13, and 24 are allowable over the cited references.

C. Dependent claims 3, 14, and 25 are Not Separately Argued

D. Dependent claims 4, 15, and 26 are Not Separately Argued

E. Dependent claims 5, 16, and 27 are Patentable over the Prior Art

Dependent claims 5, 16, and 27 depend on claims 4, 15, and 26 and further add the limitation that the information also comprises a form name. As used herein, the form name identifies the name of the form that the element was transferred from.

In rejecting these claims, the Office Action relies on Namu page 4, item "Form Name". However, it is Appellants' understanding of Namu, that the "Form Name" merely identifies the name of the HTML form that the user is creating. In this regard, the name of the form that the element was transferred from is not taught or suggested by Namu. The Office Action fails to provide any text or description of the Namu screen shots.

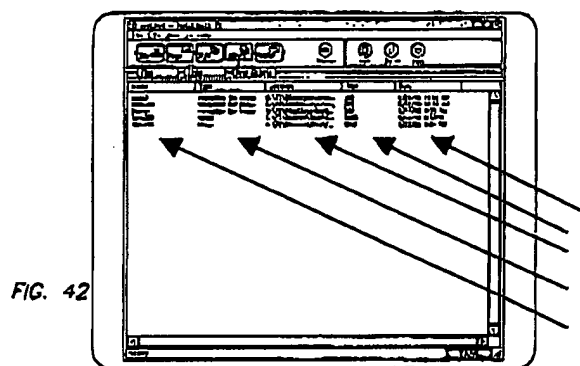
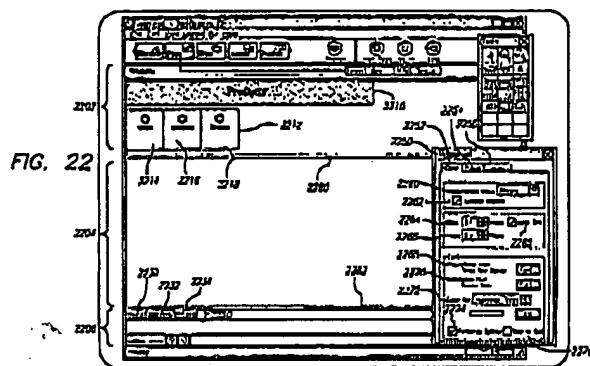
Appellants took the liberty of downloading the appropriate version of the Nano WebEditor to learn more about the cited reference. Appendix C contains 3 pages printed out from the help menus of the Nano WebEditor. As can be seen by the help menus, the Form referred to in Namu is merely an HTML Form that is created by the user and is not a form that elements were transferred from (as claimed).

Thus Appellants submit that dependent claims 5, 16, and 27 are allowable over the cited references.

F. Dependent claims 6, 17, and 28 are Patentable over the Prior Art

As described above, these dependent claims further illustrate how to display the mapping. In this regard, the element name and the name of the HTML file that the element was transferred to is displayed in a row of a table.

To teach this claim element, the Office Action relies on Arora Figure 22 item 2202 and col. 10, lines 18-40. The Office Action also relies on Arora Figure 42 and column 14, lines 33-36. However, Appellants note that Arora's Figure 22 does not show a table whatsoever. Instead, there are various graphical buttons. In addition, as indicated in the prior Office Action responses and Appal Brief, FIGS. 22 and 42 and col. 14, lines 32-36 of Arora provide for an Assets Display that shows the files, links, and objects in a data processing system.



The assets shown in FIG. 42 are the draw objects for the products page of FIG. 22. Viewing FIG. 42, it appears that the Assets Display has a name of an object/file, the type of the object, a location of the object, a size of the object, and a date (see annotation). Thus, FIG. 42 merely provides a listing of the objects within a particular project. There is no mapping indicating an HTML file where the object is located. Furthermore, there is no indication or concept anywhere in FIG. 42 that the element has been transferred from a form to an HTML page. Listing objects found in a folder is not equivalent to displaying a mapping of elements from a form to an HTML file. Further, there is no suggestion to add a mapping from any other reference to the listing. The listing is merely a listing of very generalized information about objects in a particular project and the folder location where the object is stored. In this regard, the element name and the name of the HTML file name that the element was transferred to is not displayed in a row of a table in Arora's Figure 42 or elsewhere. To contend that displaying a list of elements is equivalent to displaying particular specially obtained mapping information is without merit.

Form...

Format Menu

Gives properties of the current form or selected contents. With a Button field , you can submit the information of the form to the Web server. If you want to clear an empty form, use the Clear form option. Some options are: Form Name, Target Frame name, Action, Method and Encoding Type.

You can bring the Form Properties dialog box up by activating the context popup menu with the right mouse button and choosing Form Properties menu.

General Tab

Form Properties Dialog Box

Information

Form Name

Give the name of a form that includes form fields in it.

Target Frame

If a form is related with any target frame, give the name of the frame here.

Extended...

If you want to insert extended attributes and values to the insertion position, use this button. Give names and values of attributes in the Extended Attributes dialog box. Also, you can modify or remove extended attributes with this dialog box.

Hidden Field

If you want to insert a hidden field in the current form, use this button. Provide an attribute name and value in the Form Hidden Fields dialog box. Also, you can modify or remove a hidden form field in this dialog box.

Clear Form

Clear the current form, if you do not want to use it. Any fields in the cleared form do not belong to any form.

Settings

Action

Enter the URL of the query server to which the form contents will be submitted. If no URL is entered, the current document URL will be used.

Method

Enter the method of submitting information to the form handler: Get or Post.

The Get method encodes the form's name/value pairs and assigns the information to a server variable called QUERY_STRING.

The Post method passes the name/value pair directly to the form handler as input for the post.

Encoding Type

The standard used to encode the form data that is passed to the form handler. Leave this field blank to use the default encoding method: application/x-www-form-urlencoded.

Form Field

Insert Menu

Contains the following sub Form Field Menus: One Line Text Box , Scrolling Text Box , Check Box , Radio Button , Drop-Down Menu , Push Button and Image Form .

Various Form Fields can be sent to the form handling server by using the submit button of the Push Button Field, or canceled with the reset button.

Use the Form command in the Format menu to give common properties to a form.

Additionally, merely displaying the values of a properties window (Arora column 10, lines 61-67) (i.e., a grid of pixels, the font used and the colors, etc.) and a list of draw objects (Arora column 11 lines 7-20) (i.e., a list of objects on a page) is not even remotely similar to displaying a mapping that indicates the relationship between an element, a form (that the element was transferred from), and an HTML file (as claimed).

Thus Appellants submit that dependent claims 6, 17, and 28 are allowable over the cited references.

G. Dependent claims 7, 18, and 29 are Patentable over the Prior Art

Dependent claims 7, 18, and 29 further elaborate on the table of claims 6, 17, and 28. Specifically, these claims indicate that a table has cells that are defined by row and column and the mapping is entered into the cell. The Office Action relies on Arora Figure 39 to teach this claim element. The description of Arora's Figure 39 (col. 13, line 57 – col. 14, line 5) provides:

FIG. 39 is an example of a matrix 3902 generated in accordance with the edges and draw objects. Matrix 3902 has a number of rows equal to the number of row edges plus one. The matrix has a number of columns equal to the number of column edges plus one. Thus, in the example, the matrix has seven columns and eight rows. Each element 3904 of the matrix has a pointer field and an occupied flag. The element of row 7, column 2 of the matrix points to object number 1 and is marked as occupied. The next four elements in row 2 are also marked as occupied. The element of row 3, column 3 points to object number 2 and is marked as occupied. The element of row 2, column 5 points to object number 3 and is marked as occupied. The next four elements in column 5 are also marked as occupied. The element of row 5, column 3 points to object number 4 and is marked as occupied.

Thus, Figure 38 merely describes cells that have a pointer field and an occupied flag. In this regard, there is no mapping entered into a particular cell. Further, Arora completely fails to describe the use of such a mapping or table.

Thus Appellants submit that dependent claims 7, 18, and 29 are allowable over the cited references.

H. Dependent claims 8, 19, and 30 are Not Separately Argued

I. Dependent claims 9, 20, and 31 are Not Separately Argued

J. Dependent claims 10, 21, and 32 are Patentable over the Prior Art

Dependent claims 10, 21, and 32 further elaborate the flagging of claims 9, 20, and 31. Specifically, a project file is read and the HTML file name is extracted from the project file. Thereafter, the HTML file name is used to search for the HTML file. If the HTML file is not found, the mapping is flagged as invalid.

In rejecting this claim, the Office Action provides: "claim 10 incorporates substantially similar subject matter as claimed in claims 1 and 9, and is rejected along the same rationale."

However, claims 1 and 9 do not provide for reading a project file; claims 1 and 9 do not provide for extracting the HTML filename from the project file; claims 1 and 9 do not provide for searching for the HTML file using the extracted HTML filename; and claims 1 and 9 do not provide for flagging the mapping as invalid when the extracted HTML file is not found. In fact, every step of these independent claims are substantially different from the claims on which they depend. Lisle fails to teach each aspect of these claims. In addition, the remaining references also fail to teach the claimed elements.

Thus, Appellants submit that dependent claims 10, 21, and 32 are allowable over the cited references.

K. Dependent claims 11, 22, and 33 are Patentable over the Cited Art

These dependent claims provide for accepting a modified mapping and storing the modified mapping in a project file. In rejecting these claims, the Office Action relies on HTMLed providing various editing buttons on a tool bar for modifying and saving into an HTML file – sample.htm.

As stated above, the mapping as claimed is not taught or suggested, explicitly or implicitly, by HTMLed. Without teaching such a mapping, HTMLed cannot possibly teach a modified mapping or the storage of such a modified mapping.

In addition, Appellants submit that various editing buttons on a tool bar for modifying which elements are in an HTML form is not even remotely similar to modifying a mapping and storing the mapping in a project file.

As claimed, there is a project file AND an HTML file. The claims do not provide that these files are the same. The Office Action provides that the sample.htm file is both the HTML file and

the project file. Such an equivalence is not taught or suggested by the claims or the specification of the present invention. Further, a project file (as claimed) is not equivalent to the sample.htm file. As claimed, the project file contains information that establishes the relationship between an element that has been transferred from the form to the HTML page and the HTML file associated with the HTML page. HTMLed's sample.htm file does not provide any such information. Accordingly, HTMLed fails to teach various aspects of these dependent claims.

Thus, Appellants submit that dependent claims 11, 22, and 33 are allowable over the cited references.

IX. Conclusion

In light of the above arguments, Appellants respectfully submits that the cited references do not anticipate nor render obvious the claimed invention. More specifically, Appellants' claims recite novel physical features which patentably distinguish over any and all references under 35 U.S.C. §§ 102 and 103. As a result, a decision by the Board of Patent Appeals and Interferences reversing the Examiner and directing allowance of the pending claims in the subject application is respectfully solicited.

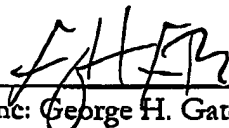
Respectfully submitted,

GATES & COOPER LLP

Attorneys for Applicants

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: October 8, 2003

By: 
Name: George H. Gates
Reg. No.: 33,500

GHG/JSF/io

G&C 30571.179-US-01

APPENDIX A

1. A method of displaying a relationship between an HTML file and an element from a form, wherein the element is in an HTML page, comprising:
 - reading information from a project file, the information comprising a relationship between the element that has been transferred from the form to the HTML page and the HTML file associated with the HTML page;
 - processing the information to map the element from the form to the HTML file; and
 - displaying the mapping on a graphical user interface that indicates the relationship between the element, the form, and the HTML file.
2. The method of claim 1, wherein the information is generated when the element is transferred from the form to the HTML page associated with the HTML file.
3. The method of claim 1, wherein the element is selected from a group comprising:
 - a visual control; and
 - a non-visual control, selected from a group comprising a button, a picklist, and a data entry box.
4. The method of claim 1, wherein the information comprises an element name and an HTML file name.
5. The method of claim 4, wherein the information further comprises a form name.
6. The method of claim 1, wherein the displaying the mapping comprises presenting an element name and an HTML file name in a row of a table.
7. The method of claim 6, wherein the table comprises cells defined by the row and the column of the cell, and the mapping is entered into a cell of the table.

8. The method of claim 1, wherein the displaying the mapping comprises presenting the element name and the HTML file name in a column of a table.

9. The method of claim 1, further comprising flagging an invalid mapping between the element and the HTML file.

10. The method of claim 9, wherein the flagging an invalid mapping between the element and the HTML file comprises:

- reading the project file;
- extracting the HTML filename from the project file;
- searching for the HTML file using the extracted HTML filename; and
- flagging the mapping as invalid when the extracted HTML file is not found.

11. The method of claim 1, further comprising:
accepting a modified mapping; and
storing the modified mapping in the project file.

12. A apparatus for displaying a relationship between an HTML file and an element from a form, wherein the element is in an HTML page, comprising:

- means for reading information from a project file, the information comprising a relationship between the element that has been transferred from the form to the HTML page and the HTML file associated with the HTML page;

- means for processing the information to map the element from the form to the HTML file;
- and

- a display for presenting the mapping to a user on a graphical user interface that indicates the relationship between the element, the form, and the HTML file.

13. The apparatus of claim 12, wherein the information is generated when the element is transferred from the form to the HTML page associated with the HTML file.

14. The apparatus of claim 12, wherein the element is selected from a group comprising:
a visual control;
a non-visual control, selected from a group comprising a button, a picklist, and a data entry
box.
15. The apparatus of claim 12, wherein the information comprises an element name and
an HTML file name.
16. The apparatus of claim 15, wherein the information further comprises a form name.
17. The apparatus of claim 12, wherein the means for displaying the mapping comprises
means for presenting the element name and the HTML file name in a row of a table.
18. The apparatus of claim 17, wherein the table comprises cells defined by the row and
the column of the cell, and the mapping is entered into a cell of the table.
19. The apparatus of claim 12, wherein the means for displaying the mapping comprises
means for presenting the element name and the HTML file name in a column of a table.
20. The apparatus of claim 12, further comprising means for flagging an invalid mapping
between the element and the HTML file.
21. The apparatus of claim 20, wherein the means for flagging an invalid mapping
between the element and the HTML file comprises:
means for reading the project file;
means for extracting the HTML filename from the project file;
means for searching for the HTML file using the extracted HTML filename; and
means for flagging the mapping as invalid when the extracted HTML file is not found.
22. The apparatus of claim 12, further comprising:
means for accepting a modified mapping; and

means for storing the modified mapping in the project file.

23. An article of manufacture, embodying logic to perform a method of displaying a relationship between an HTML file and an element that has been transferred from a form to an HTML page, the method comprising:

reading information from a project file, the information comprising a relationship between an element that has been transferred from a form to an HTML page and the HTML file associated with the HTML page;

processing the information to map the element from the form to the HTML file; and

displaying the mapping on a graphical user interface that indicates the relationship between the element, the form, and the HTML file.

24. The article of manufacture of claim 23, wherein the information is generated when the element is transferred from the form to the HTML page associated with the HTML file.

25. The article of manufacture of claim 23, wherein the element is selected from a group comprising:

a visual control; and

a non-visual control, selected from a group comprising a button, a picklist, and a data entry box.

26. The article of manufacture of claim 23, wherein the information comprises an element name and an HTML file name.

27. The article of manufacture of claim 26, wherein the information further comprises a form name.

28. The article of manufacture of claim 23, wherein the displaying the mapping comprises presenting the element name and the HTML file name in a row of a table.

29. The article of manufacture of claim 28, wherein the table comprises cells defined by the row and the column of the cell, and the mapping is entered into a cell of the table.

30. The article of manufacture of claim 23, wherein the displaying the mapping comprises presenting the element name and the HTML file name in a column of a table.

31. The article of manufacture of claim 23, wherein the method further comprises flagging an invalid mapping between the element and the HTML file.

32. The article of manufacture of claim 31, wherein the flagging an invalid mapping between the element and the HTML file comprises:

reading the project file;

extracting the HTML filename from the project file;

searching for the HTML file using the extracted HTML filename; and

flagging the mapping as invalid when the extracted HTML file is not found.

33. The article of manufacture of claim 23, wherein the method further comprises:

accepting a modified mapping; and

storing the modified mapping in the project file.

34. A computer readable data structure for representing a software project in a single file, the software project comprising a project application defined by executable programming logic, and a project environment for developing the application, the data structure comprising:

a first section comprising the executable programming logic needed to load and execute the project application in the computer; and

a second section for storing data required to restore the project environment, and for storing information comprising a relationship between elements that have been transferred from a form to an HTML page and HTML files associated with the HTML page in the project;

wherein the relationship between elements, the form, and the HTML file is displayed in a graphical user interface.

NATIONAL BESTSELLER!

TEACH YOURSELF WEB PUBLISHING with HTML IN A WEEK

Learn to design
and create
pages on the
Internet's World
Wide Web
with HTML

Find out how to
link your pages
to each other
and to other
Web sites—
anywhere in
the world

Discover how to
add images,
sound, video,
and forms to
your Web
documents

"Teach Yourself Web Publishing with
HTML in a Week is the most complete
HTML reference I have seen... a
must-have reference for anyone
developing HTML."

Bruce Morris
Web Developer's Journal

Covers the
Netscape
extensions
to HTML

Laura Lemay

26

28
SAMS
PUBLISHING

Copyright © 1995 by Sams Publishing
FIRST EDITION

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. For information, address Sams Publishing, 201 W. 103rd St., Indianapolis, IN 46290.

International Standard Book Number: 0-672-30667-0

Library of Congress Catalog Card Number: 94-69281

98 97 96 95 4

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit, the number of the book's printing. For example, a printing code of 95-1 shows that the first printing of the book occurred in 1995.

*Composed in 11/13 Garamond and MCPdigital by Macmillan Computer Publishing
Printed in the United States of America*

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

13

Forms and Image Maps

Summary

Forms and image maps are two of the most interesting innovations that have been added in recent times to Web publishing. With the advent of these features, the Web changed from being a publishing medium with hypertext links to a fully interactive environment with the potential for being something entirely new.

In this chapter, you've learned how to lay out form elements in HTML as well as how to process the results on the server when that form is submitted. You should now know the difference between GET and POST and be able to explain why one is better. And you know the difference between a CERN and NCSA image map, how to create both of them, and how to connect clickable images, map files, and gateway scripts on the appropriate servers.

In addition, you've learned about the remaining tags defined by the current version of HTML—those that define the elements of a form. Table 13.1 presents a quick summary of all the tags and attributes you've learned about in this chapter.

Table 13.1. HTML Tags from Chapter 13.

Tag	Use
<code><FORM>...</FORM></code>	A form. You can have multiple forms within a document, but forms cannot be nested.
METHOD	An attribute of the <code><FORM></code> tag, indicating the method with which the form input is given to the script that processes the form. Possible values are GET and POST.
ACTION	An attribute of the <code><FORM></code> tag indicating the script to process the form input. Contains a relative path or URL to the script.
<code><INPUT></code>	A form element.
TYPE	An attribute of the <code><INPUT></code> tag indicating the type of form element. Possible values are SUBMIT, RESET, TEXT, RADIO, CHECKBOX and HIDDEN. SUBMIT creates a button to submit the form to the script which processes the input. RESET creates a button which resets the default values of the form, if any. TEXT creates a single-line text field. RADIO creates a radio button.



Tag	Use
	CHECKBOX creates a check box
	HIDDEN creates a form element that is not presented but has a name and a value that can then be passed onto the script that processes the form input.
NAME	An attribute of the <INPUT>, <SELECT>, and <TEXTAREA> tags. Indicates the name of the variable which holds the eventual value of this element, as submitted to the script.
VALUE	An attribute of the <INPUT> tag, indicating the default value for the form element, if any, or the value submitted with the NAME to the script. For SUBMIT and RESET buttons, VALUE indicates the label of the button.
SIZE	An attribute of the <INPUT> tag used only when TYPE is TEXT. Indicates the size of the text field, in characters.
MAXLENGTH	An attribute of the <INPUT> tag used only when TYPE is TEXT. Indicates the maximum number of characters this text field will accept.
CHECKED	An attribute of the <INPUT> tag used only when TYPE is CHECKBOX or RADIO. Indicates that this element is selected by default.
<SELECT>	A menu or scrolling list of items. Individual items are indicated by the <OPTION> tag.
MULTIPLE	An attribute of the <SELECT> tag indicating that multiple items in the list can be selected.
SIZE	An attribute of the <SELECT> tag which causes the list of items to be displayed as a scrolling list with the number of items indicated by SIZE visible.
<OPTION>	Individual items within a <SELECT> element.
SELECTED	An attribute of the <OPTION> tag indicating that this item is selected by default.
<TEXTAREA>	A text-entry field with multiple lines.
ROWS	An attribute of the <TEXTAREA> tag indicating the height of the text field, in rows.
COLS	An attribute of the <TEXTAREA> tag indicating the width of the text field, in characters.